

MINI PROJET

CHAPITRE N°2

GRAPHES ET LABYRINTHES

*Le fichier python devra être déposé sur e-campus avant le **17 janvier 23h59**.
Un fichier texte readme n'est pas essentiel, mais il faudra alors que les docstrings et tests soient suffisamment explicites dans le fichier python.*

1 Générer un Labyrinthe

Votre code Python doit permettre de générer aléatoirement un graphe correspondant à un labyrinthe, et le représenter soit en console soit à l'aide d'une interface graphique.

Pour générer un tel graphe, on peut partir d'un graphe où les sommets sont les cases du labyrinthe et aucun arc n'est construit (tous les murs entourant chaque case sont construits). On considère ensuite l'algorithme suivant :

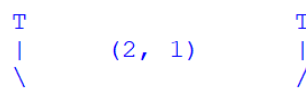
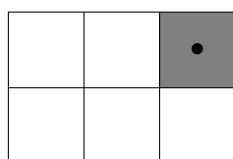
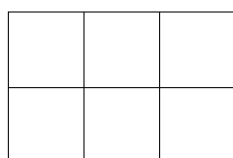
- On part de la case en position $(i; j)$ en tirant deux entiers i et j au hasard (en faisant attention à ce que la case $(i; j)$ soit bien dans le labyrinthe), on stocke dans une pile cette position et la case de départ est considérée comme visitée
- Notre position dans le labyrinthe correspond à l'élément au sommet de la pile.
- Tant que la pile n'est pas vide, on étudie les directions vers lesquelles il est possible de se déplacer (les cases encore non visitées autour de notre position), on choisit au hasard la case où on se déplace, on considère cette nouvelle case comme visitée et on construit les arcs qui relient notre position précédente à notre position actuelle (on détruit les murs nécessaires au déplacement), et on empile cette nouvelle position.
- Une position est supprimée de la pile uniquement lorsque les directions possibles pour un prochain déplacement sont au nombre de 0 ou 1.

Voici un exemple d'exécution de cet algorithme pour un labyrinthe de longueur $n = 3$ et de largeur $m = 2$. Sur cet exemple, la case en bas à gauche et numérotée $(0, 0)$, mais vous pouvez choisir les étiquettes des noeuds du graphes comme bon vous semble.

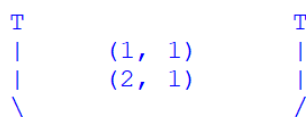
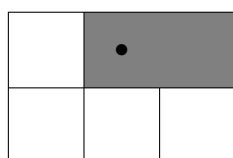
Une case blanche représente une case non visitée.

Une case grise représente une case visitée.

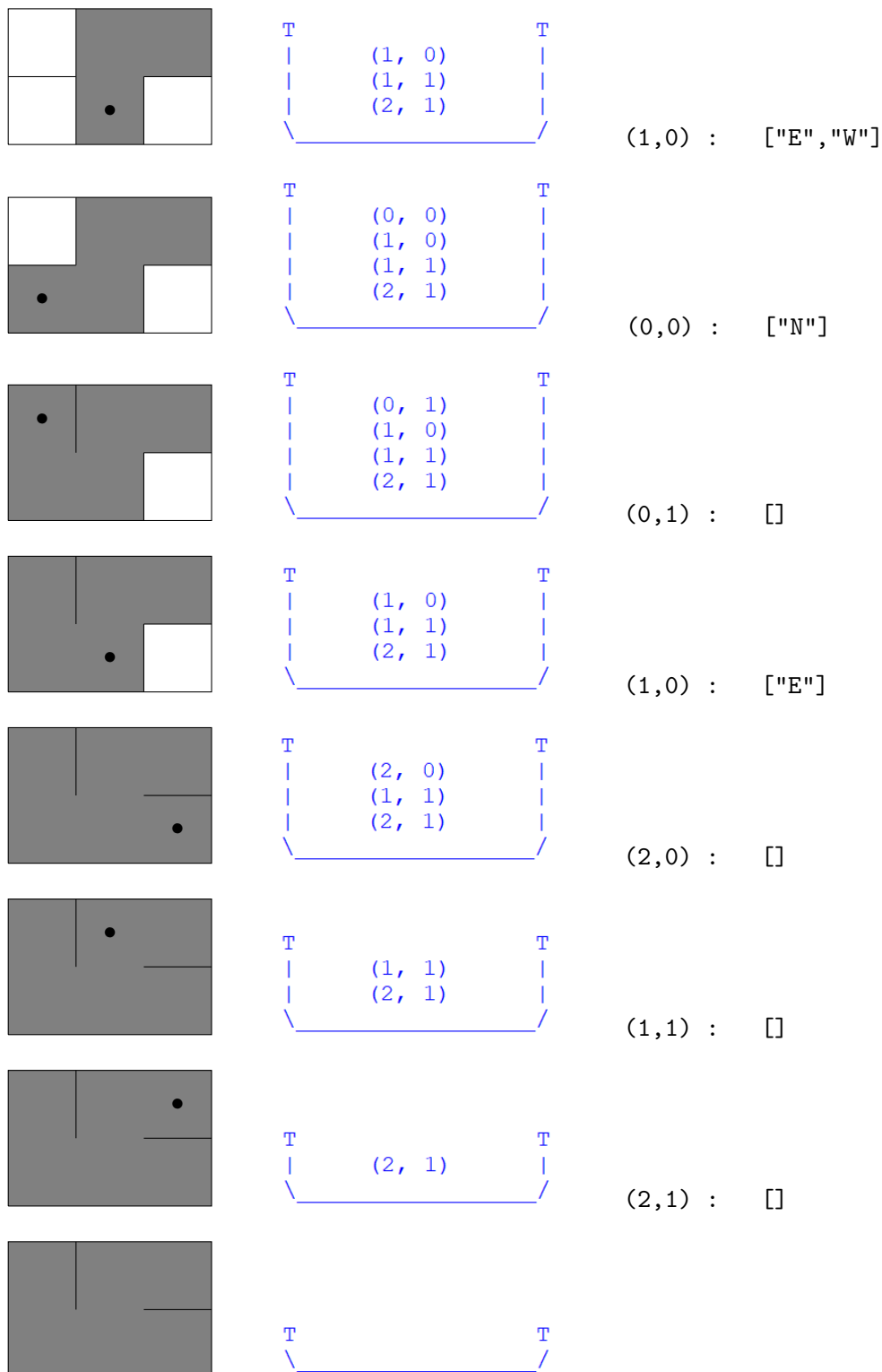
Le point représente notre position dans le labyrinthe.



$(2, 1) : ["W", "S"]$



$(1, 1) : ["W", "S"]$



2 - Construction d'une solution

A l'aide d'un parcours du graphe correspondant au labyrinthe, vous devez pouvoir proposer une solution : il doit donc y avoir la possibilité d'afficher sur le labyrinthe un trajet permettant de relier l'entrée et la sortie de labyrinthe. Ces deux dernières peuvent être placées où vous le souhaitez, par exemple entrée en haut à gauche et sortie en bas à droite.